Software Engineering

Name: Gagan Deep Singh (17BCI0140)

Team mate: Onkar Sharma (17BCE0276)

Preface

This document presents the Software Design Specification for the Weather forecasting software project by Gagan Deep Singh. The major sections of the document address the system decomposition by module, concurrent process, and data entity. The system dependencies are also described.

Section 2, Decomposition Description, gives a view of the whole system design including concurrent processes and data entities that are common amongst all system modules. This discussion includes a UML Class Diagram that depicts the entire system.

Section 4, Interface Description, goes into detail about the user interface for each module of the weather forecasting software. This is followed by an important discussion of the processes implemented in logic for each module of the system.

Section 5, Detailed Design, extends the design discussion found in Section 2 and describes the design for each system module in more detail. A UML Class diagram is included for each module design discussion. This is followed by a description of the data requirements for each module and the design of those data elements.

Contents

1. Introduction

1.1 Purpose

The purpose of the Software Design Specification is to describe the specific design of the weather forecasting software project that are to be met by the implementation effort of the team.

The design specification includes an overview of the design along with software module decomposition.

This document provides a detailed description of each software module's design. For each module, a class diagram design is given. As well, a process description is described for each module. It is in the process description that the details of what logic will need to be implemented are given.

1.2 Scope

It is within the scope of the Software Design Specification to describe the specific system design of the weather forecasting software project. This would include user interface design, object-oriented class design, process design, and data design. Any specific detail that is needed about the standards or technology used to design the software are within the scope of this document.

It is outside the scope of this document to describe in any detail at all how certain mentioned standards or technologies work and operate.

1.3 System Architecture

The system architecture chosen for the weather forecasting software is Repository Model. As large amount of data is needed to change. The advantages of this model are that it need not to consider how data is produced and sharing model is published as repository schema. Here subsystems agree upon repository model.

The Client-Server architecture is not used because there are almost no completely independent functionalities and each functionality is not large enough to require its own server. Also, this project requires data sharing between the modules which is not possible with the client server model.

The Abstract Machine model is not suited because there is no requirement of dividing the project into layers. Also, there is not requirement of incremental development of the layers. So, this model becomes difficult to follow in the project.

The Pipe and Filter model is not a good choice for the project because the project does not have a large amount of parts and they are not reusable. So, using this model will not help in any way.

1.4 Control Model

We will be using the Call and Return model which comes in the centralized control model. One subsystem is designated as the controller which manages the execution of the other subsystems. The execution is sequential. The main subsystem calls a subsystem to perform a function and after execution the control is returned to the main subsystem.

This is the best-suited model for our project because, it is a small-scale project which does not require large functionalities. So, sequential execution will give the best performance.

2. Decomposition Description

2.1 Module Decomposition

The Weather Forecasting Software has been decomposed into the following modules.

- Administrator Module: The Administrator user will be computer literate and technically competent in performing administration on computer systems. He has most power upon data. He can add, delete and modify data. He can view activity of different users contributing to the system.
- User Module: The user is anyone who is using the system for general prediction of weather data at a particular location. He has a small role in adding small piece of data to the database after verification by the concerned administrator.

The entities or modules mentioned above can access the features given to them in the system, where they have to login with their own username and password. More information is provided in SRS document.

2.2 Concurrent Process Decomposition

The Project consists of two major components, the admin and the user. A complete view of the project suggests that there are 3 processes, get prediction, add/delete data and modify data.

2.3 Data Decomposition

The data components are: Addition process and deletion or modification process.

- Addition Process: The addition process involves initialization of tables in database or adding new data in database.
- Deletion or modification Process: The process involves manipulating tables in database by deletion or modification of data.

3. Dependency Description

3.1 Inter-module Dependencies

3.1.1 Independent Modules

The following modules are independent and do not rely on any other modules to initiate them or to provide data:

- Input current parameters
- Add/Delete/Modify data

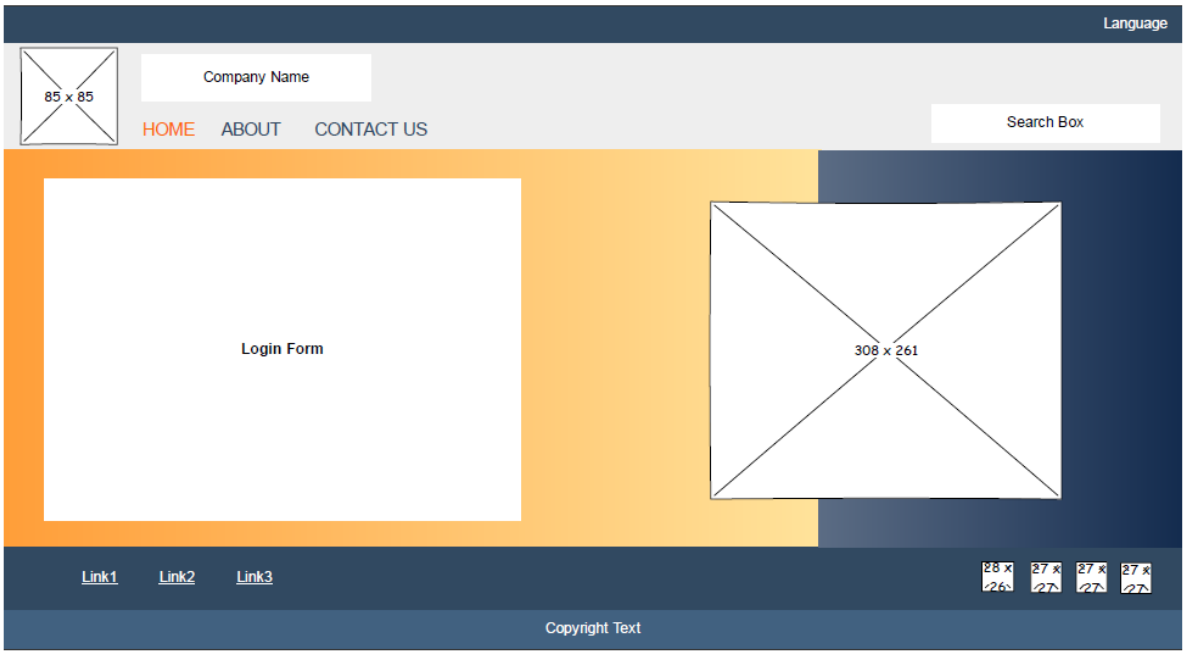
3.1.2 Dependent Modules

Get prediction takes data from the above 3 modules: Weather data like temperature, pressure and wind.

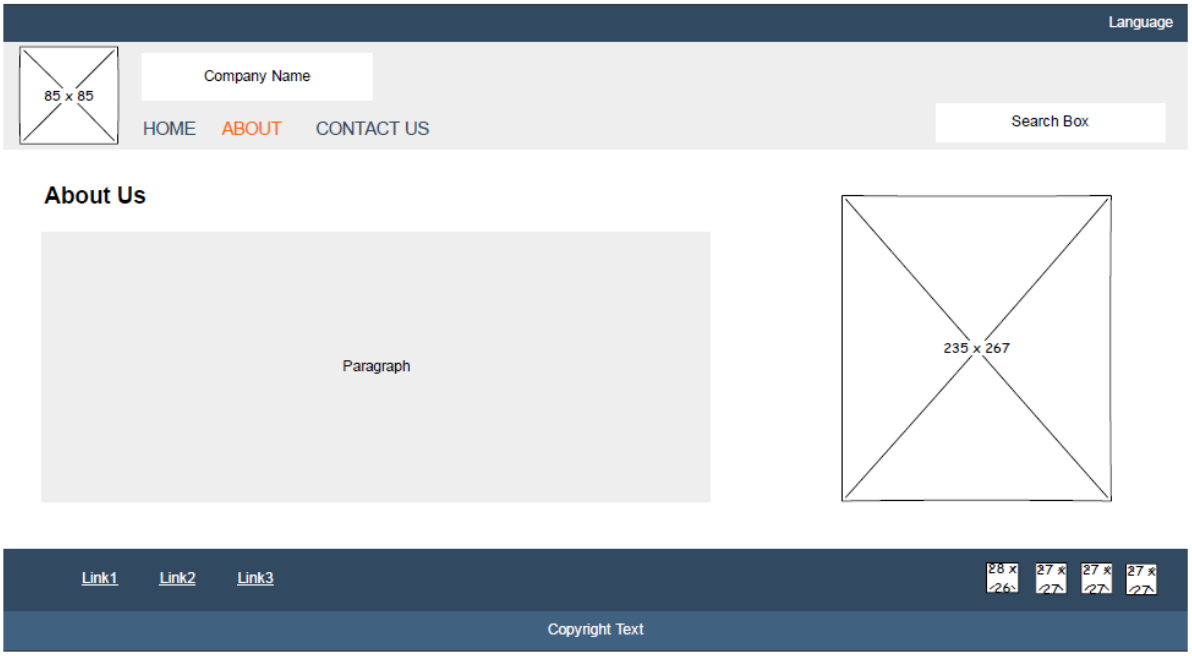
4. User Interface Description

4.1 User Interface

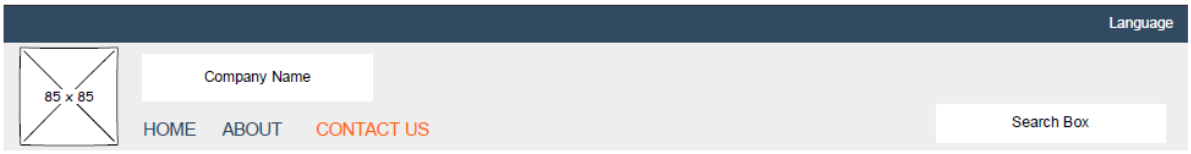
Landing Page



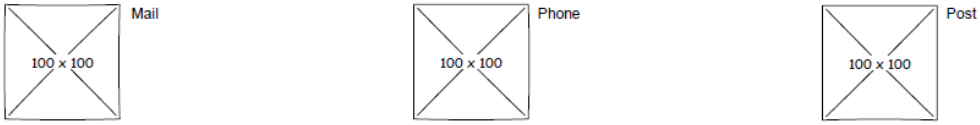
About

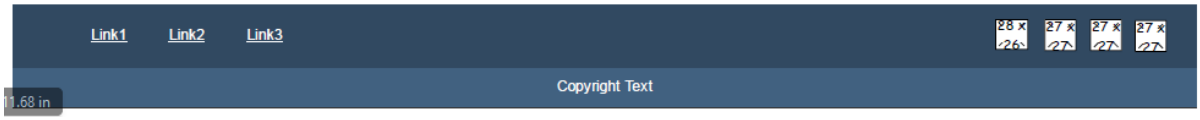


Contact

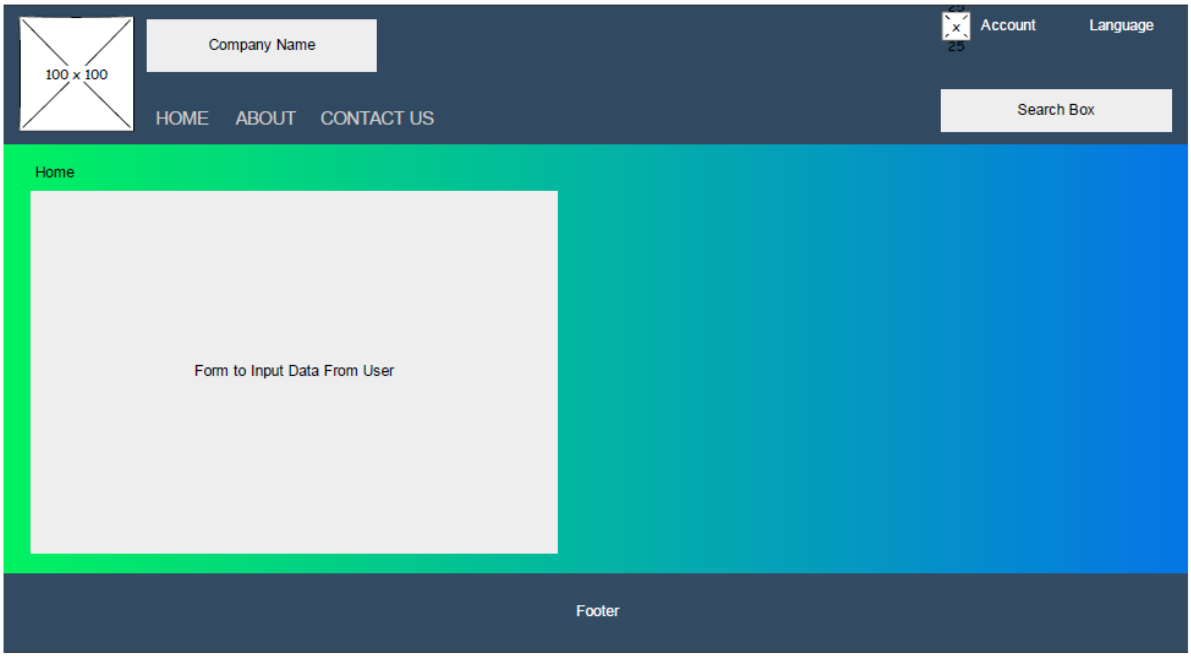


Contact Us

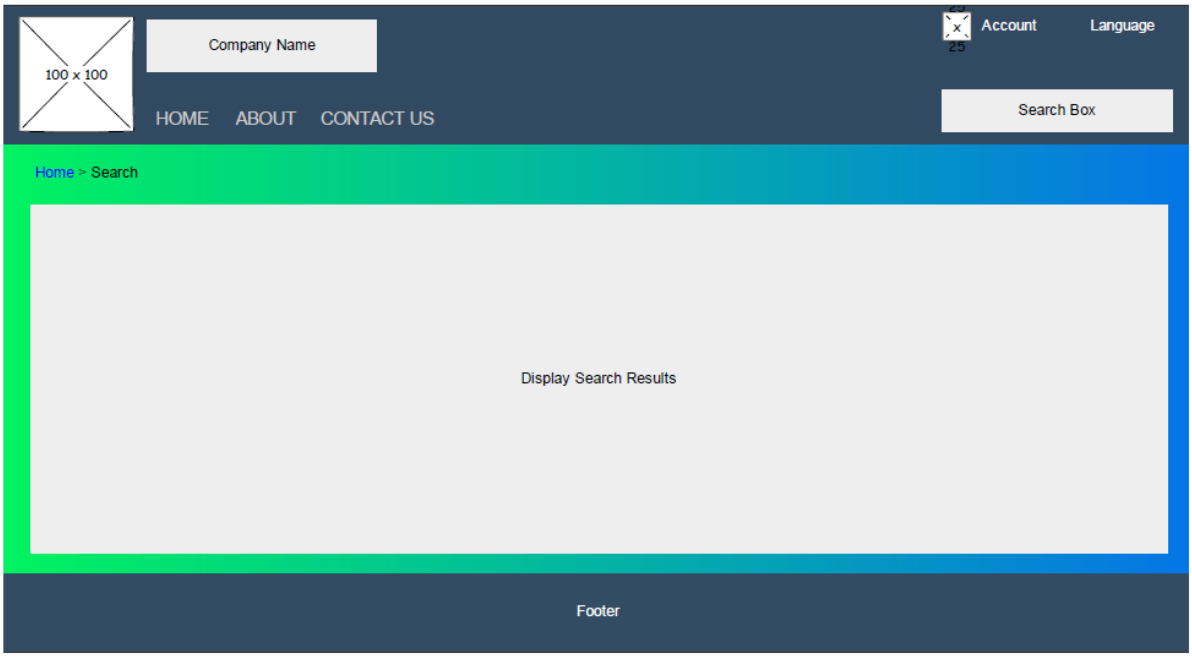




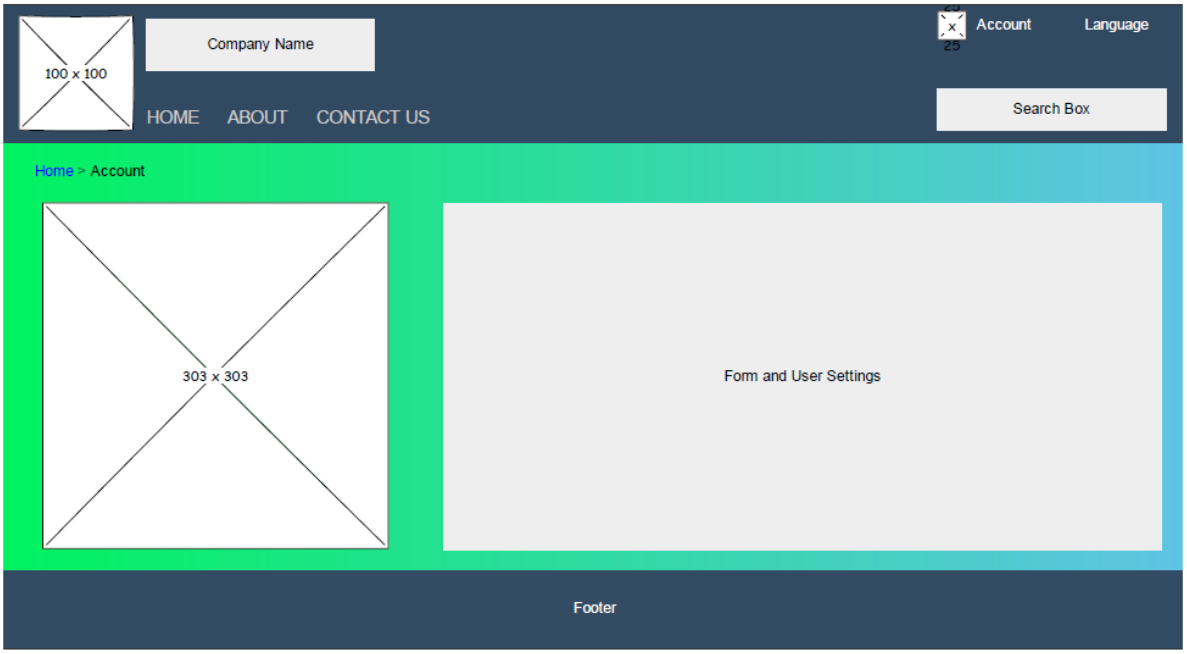
After Login



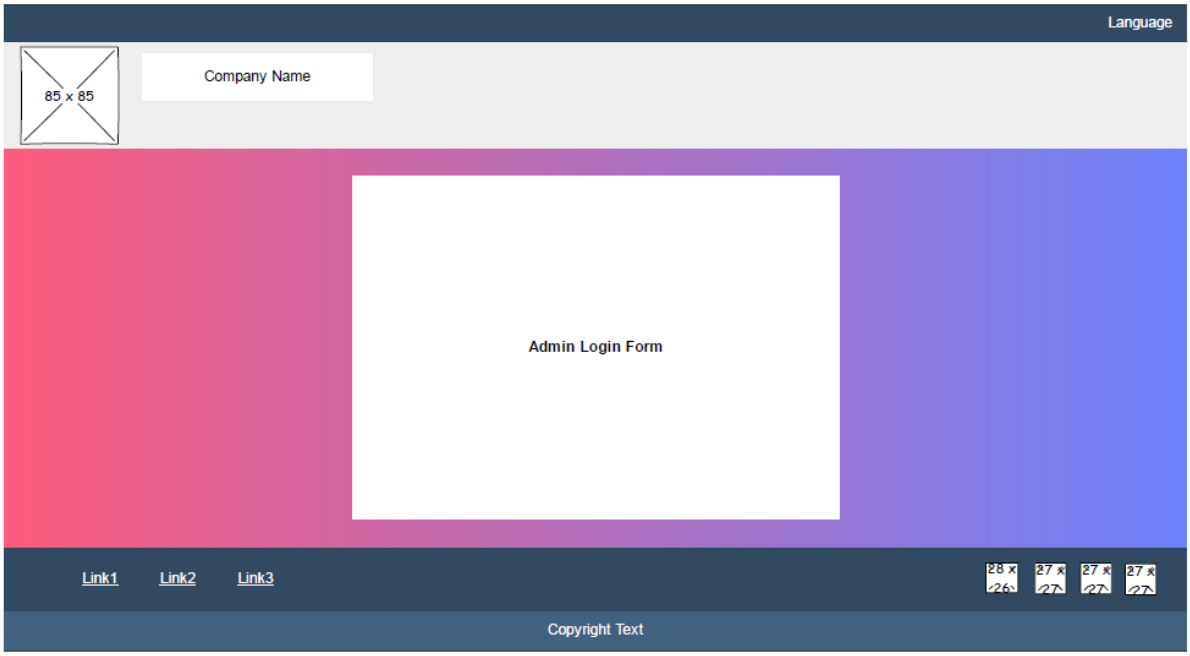
Search Display



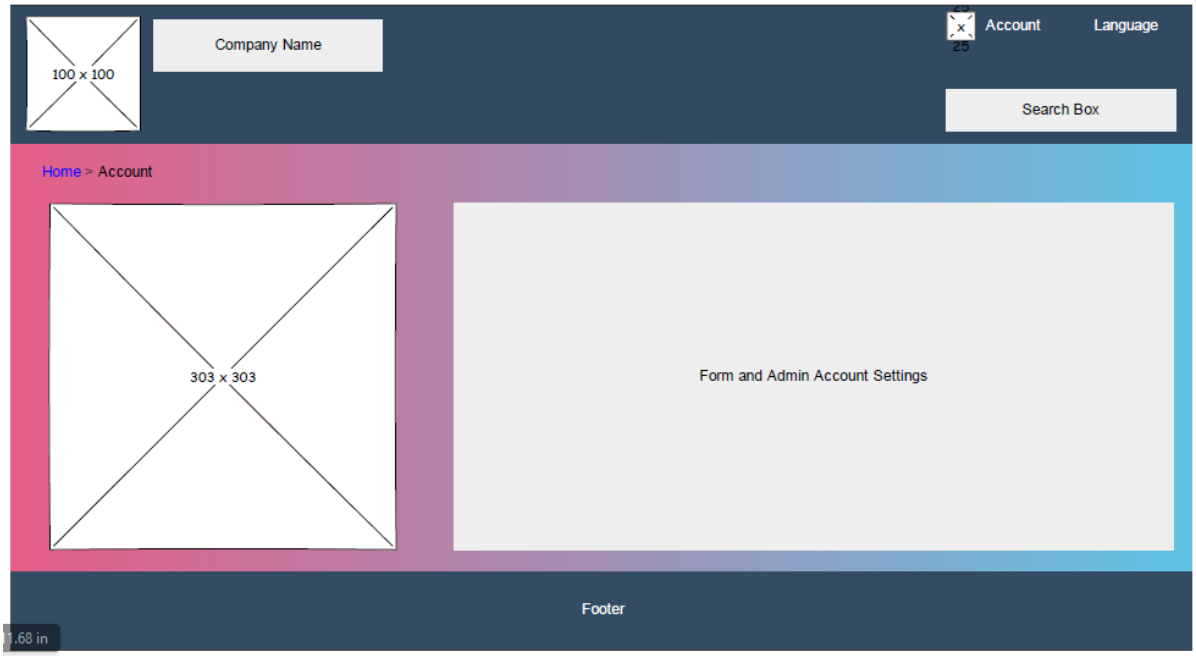
User Account



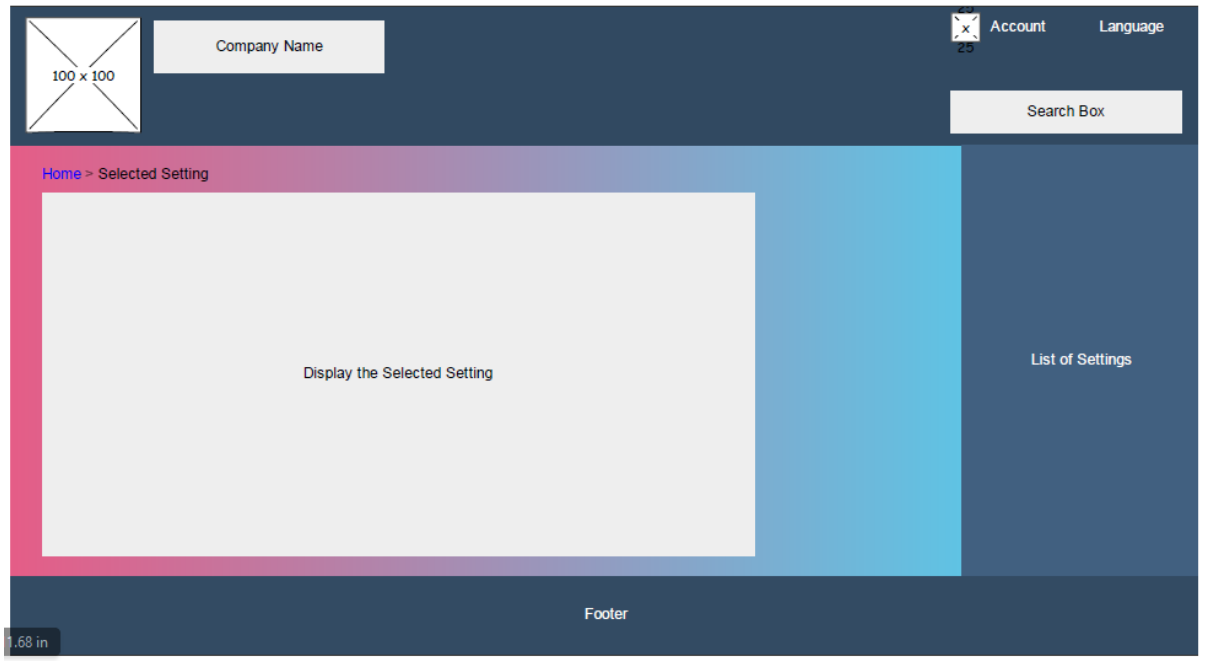
Admin Login



Admin Account



Admin After Login



4.2 Interface Description

- User logs in
- Inputs parameters
- Gets predictions
- User logs out

- Admin logs in
- Inputs parameters
- Gets predictions
- Inputs bulk data
- Adds/deletes/modifies data
- Admin logs out

5. Detailed Design Description

Refer to Appendix A: UML Class Diagram for viewing the detailed design of the system.

5.1 Admin Class:

The Admin class is capable of adding or removing data from the database. If some conditions satisfy then the admin can also modify data. This class responsible for managing the data. The admin class is also the user having many of the mentioned privileges, this shows the polymorphism character of the admin class.

5.2 User Class:

The user class can access general specified methods of GetPredictions class which let the user to view certain set of the data. User also has some role in modifying the database by just adding small amount of data available to him/her.

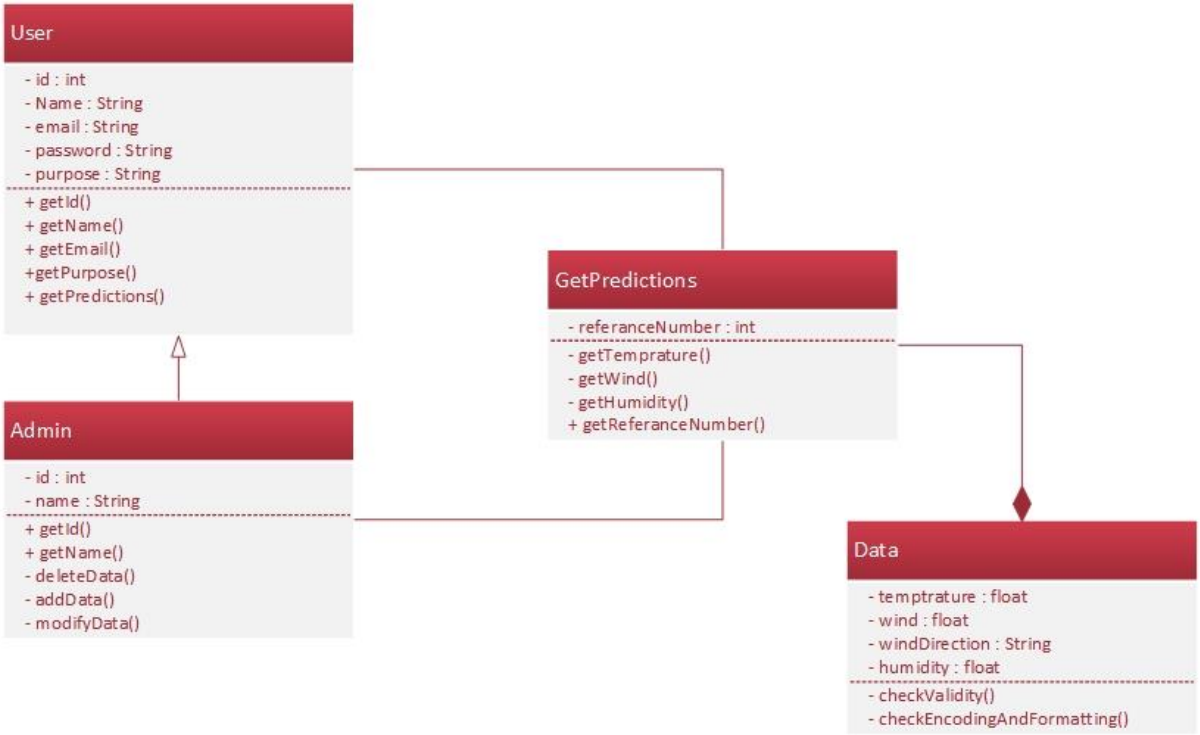
5.3 GetPredictions Class:

This class has access to all the weather data which is used to get general predictions. This class has most of attributes private so as to maintain the security of the system as anyone can extend this class.

5.4 Data Class:

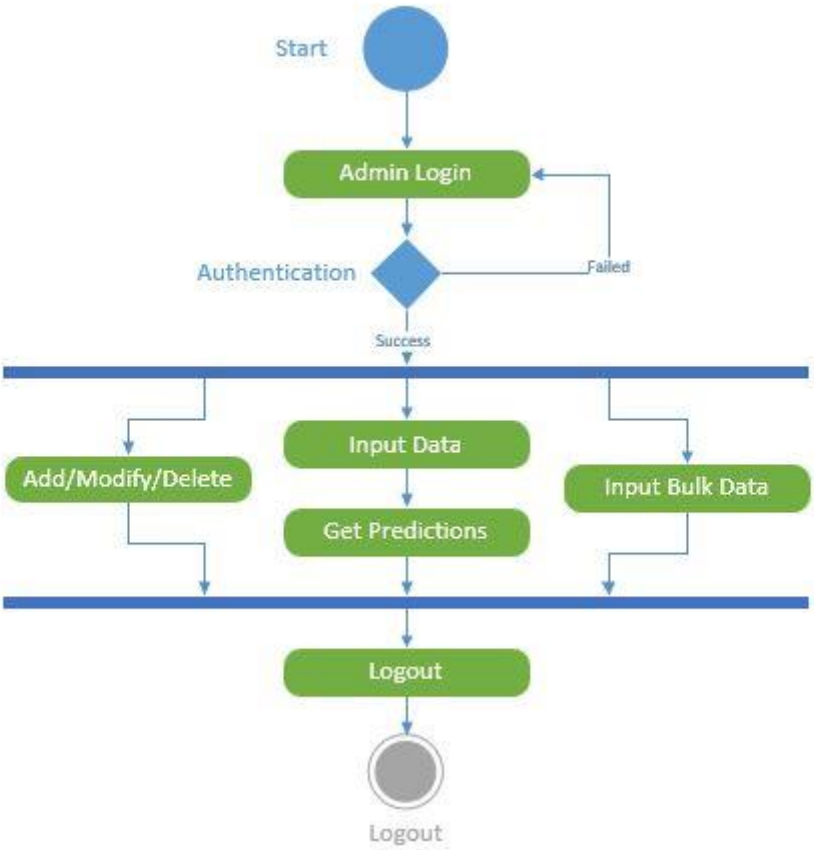
This class maintain the general data of the users, admin and the data required for weather predictions. This class has role in admin and user verification and in checking correctness of predictions.

Appendix – A: UML Class Diagram:

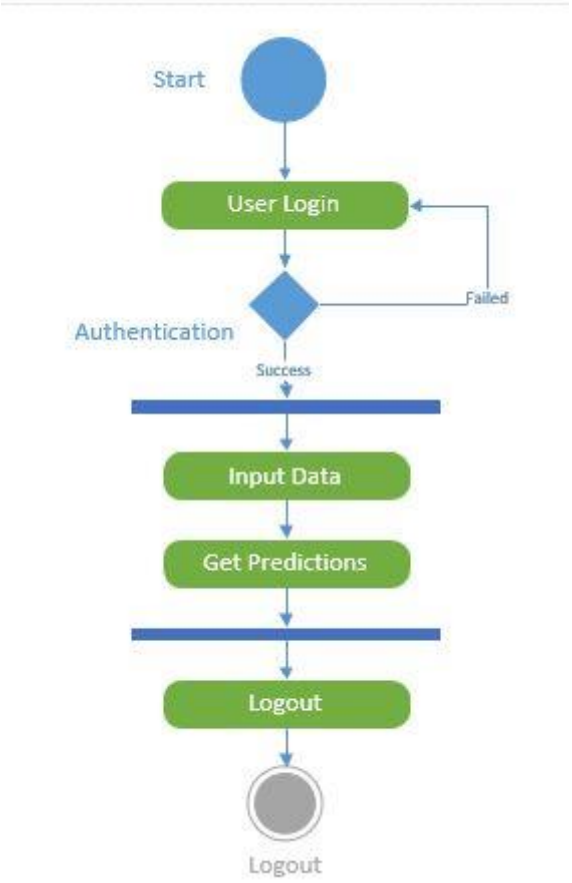


Appendix – B: Activity Diagram

a. Admin:

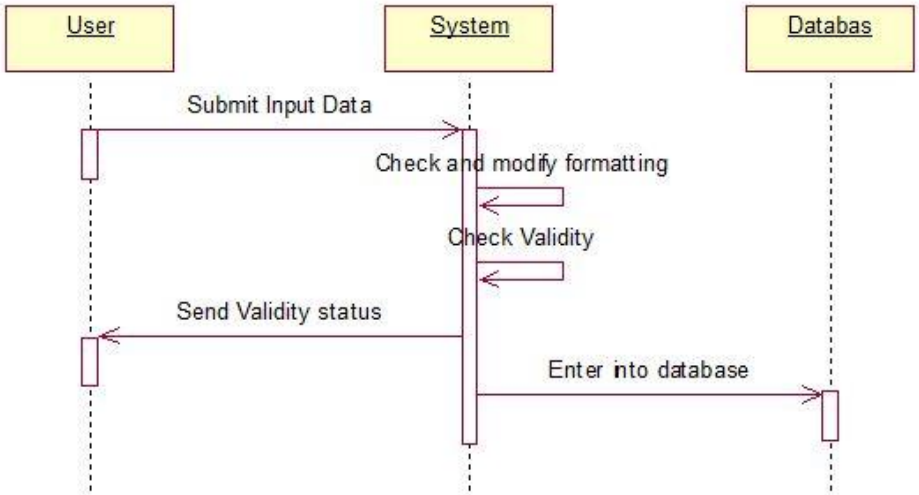


b. User:

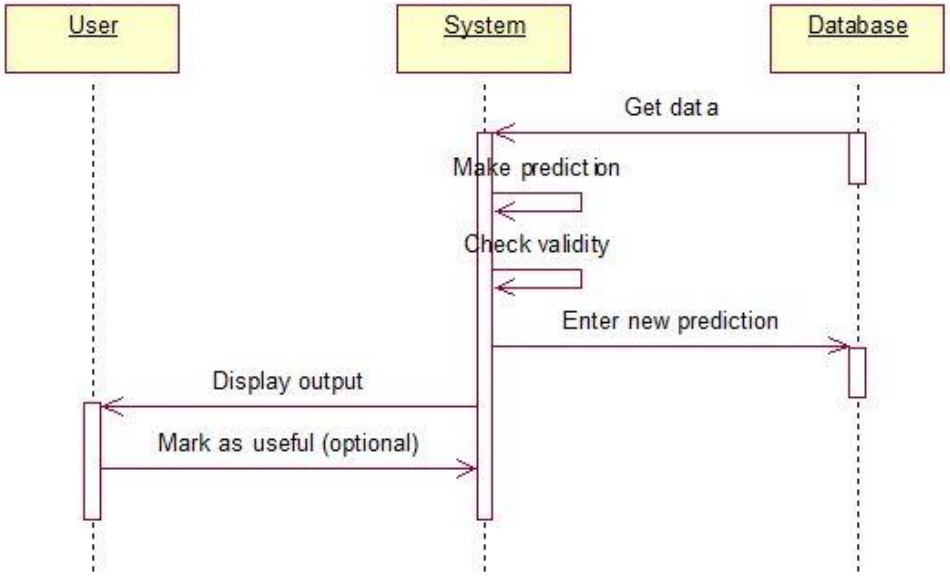

Start
User Login
Failed
Authentication
Success
Input Data
Get Predictions
Logout
Logout

Appendix C: Sequence Diagram:

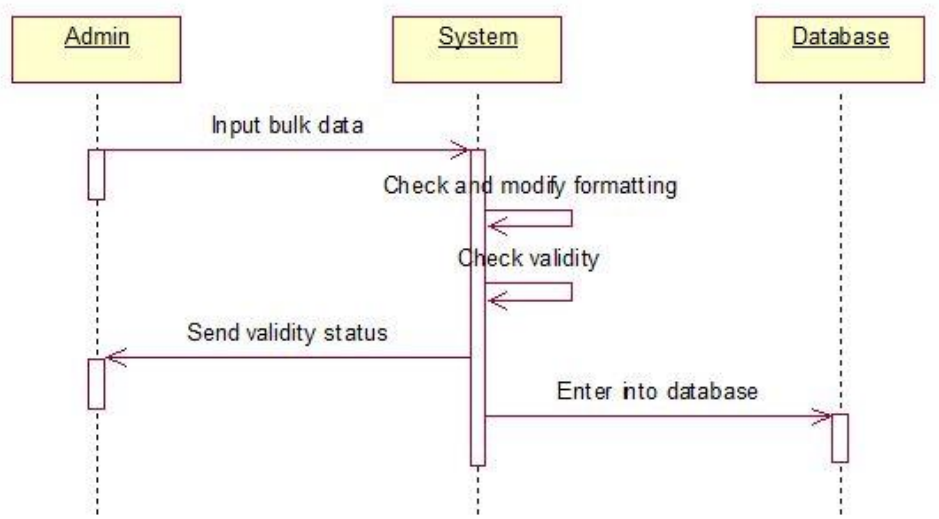
Input Current Parameters:



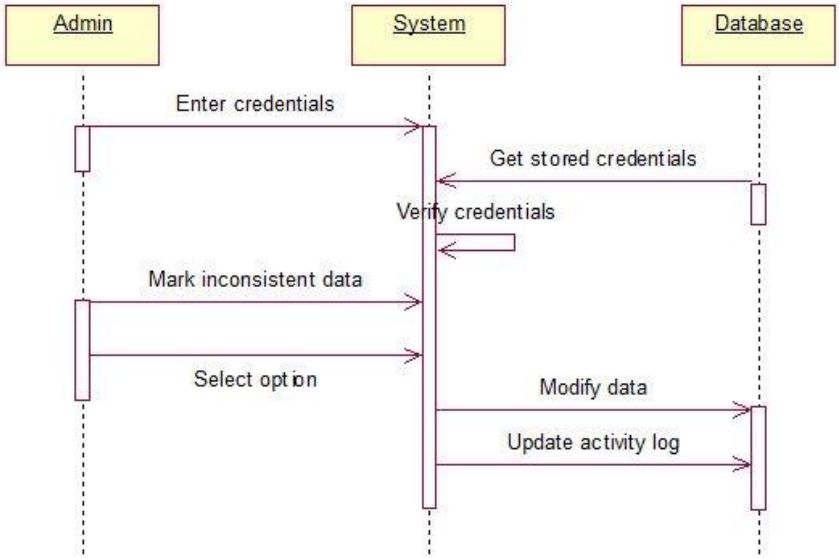
Get Predictions:



Input Bulk Data:

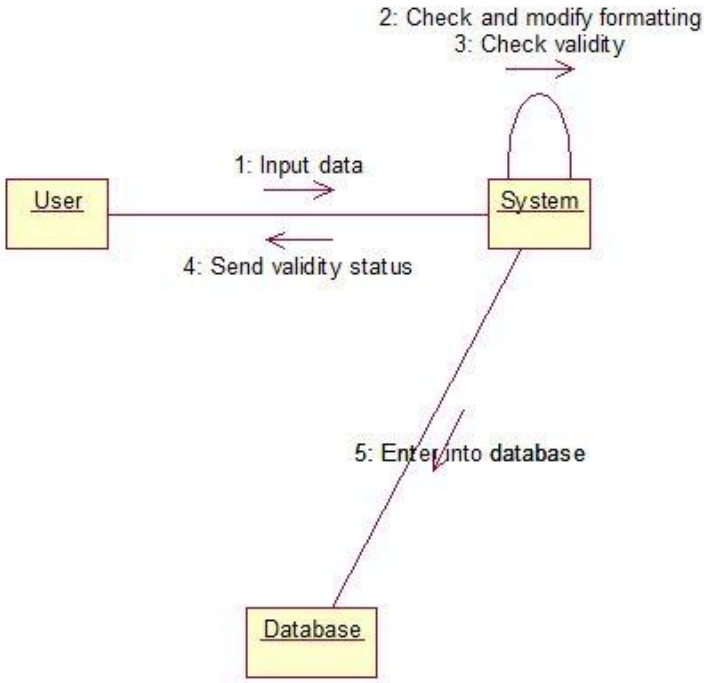


Add/Delete/Modify Data:

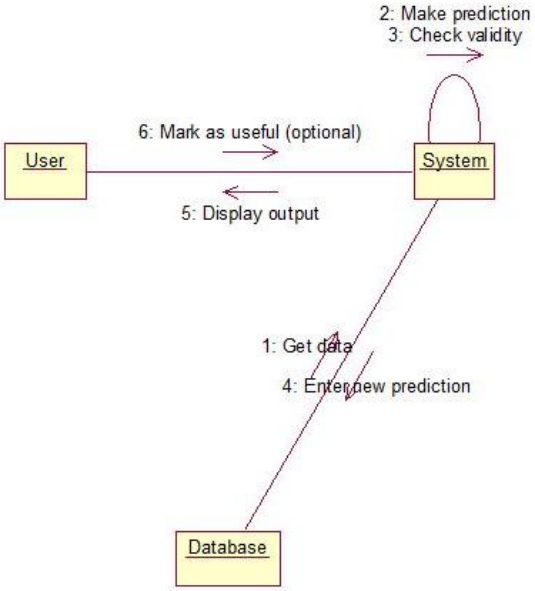


Appendix D: Collaboration diagram:

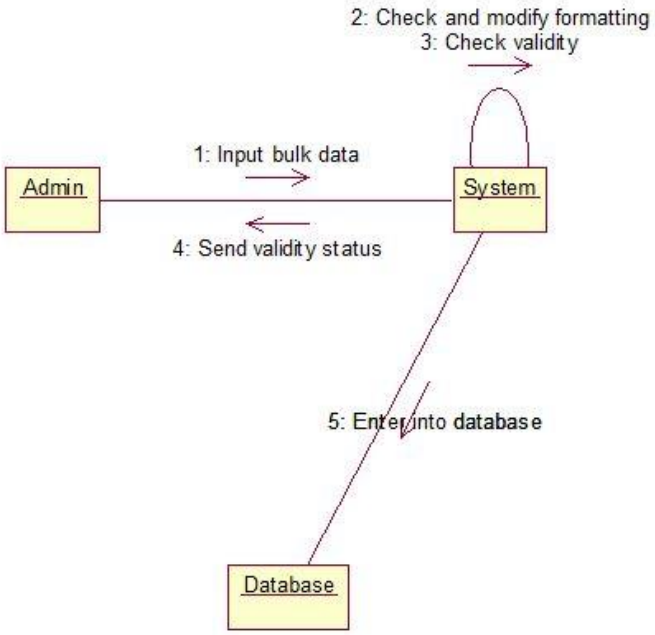
Input Current Parameters:



Get Predictions:



Input Bulk Data:



Add/Delete/Modify Data: